

50

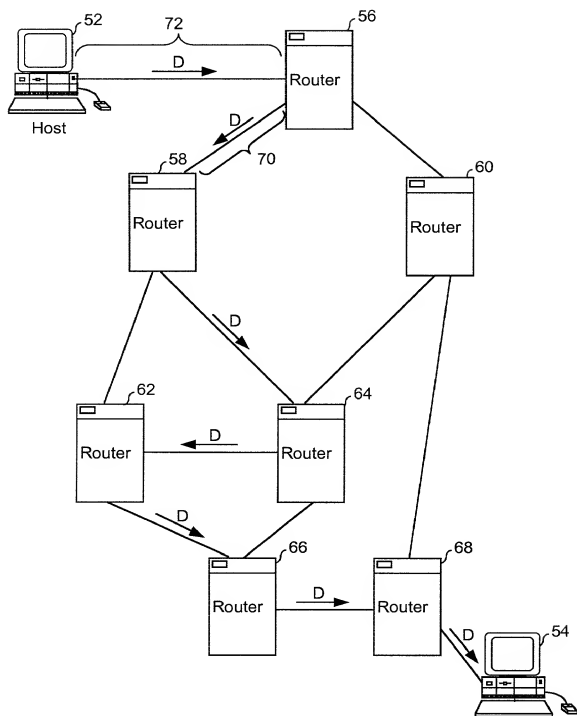
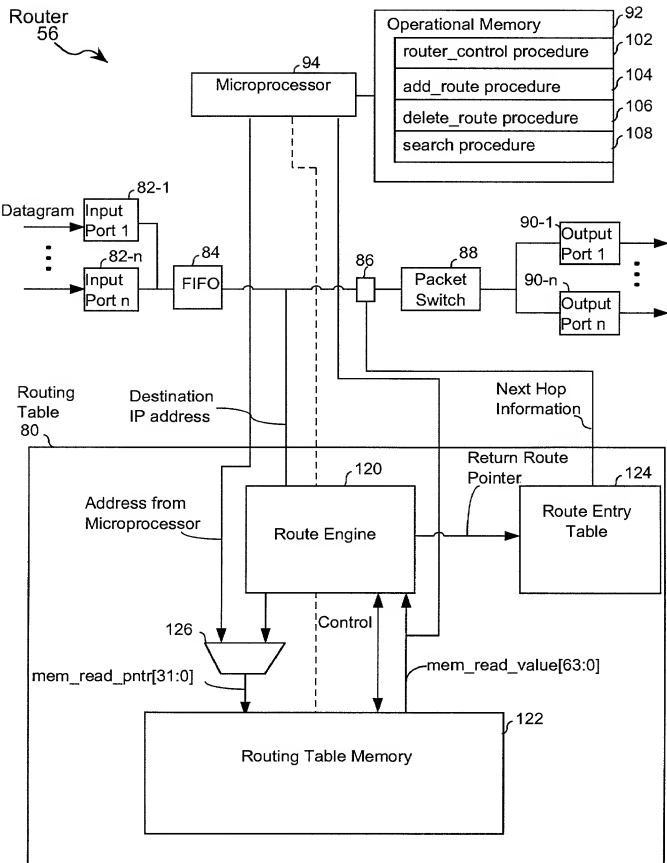


FIG. 2



Router

FIG. 3

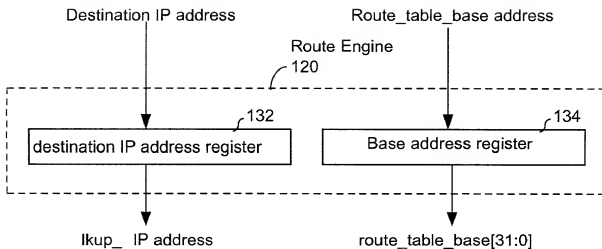


FIG. 4

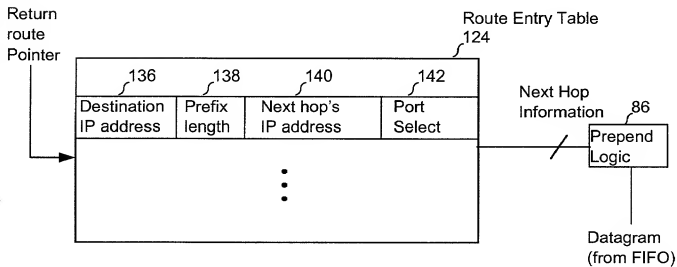
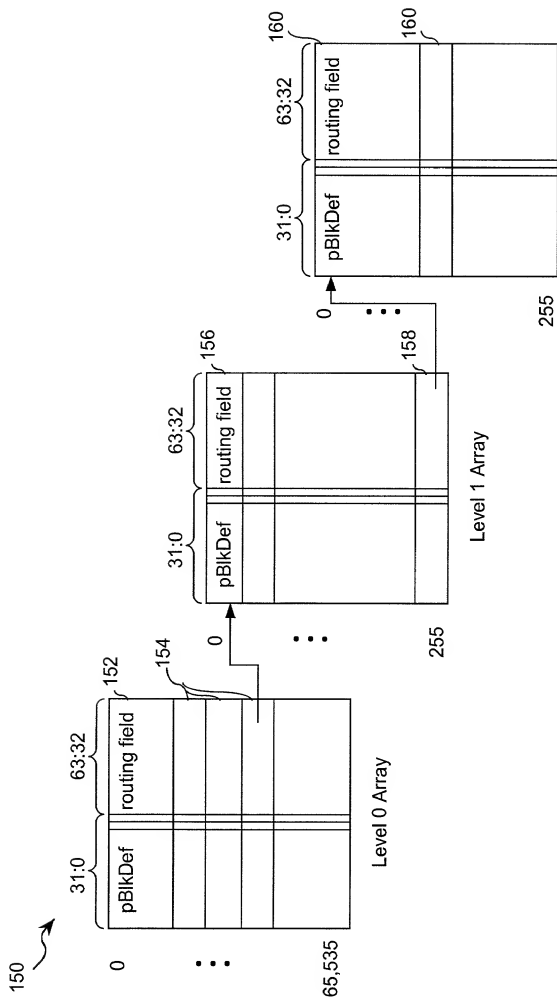


FIG. 5



170 ↗

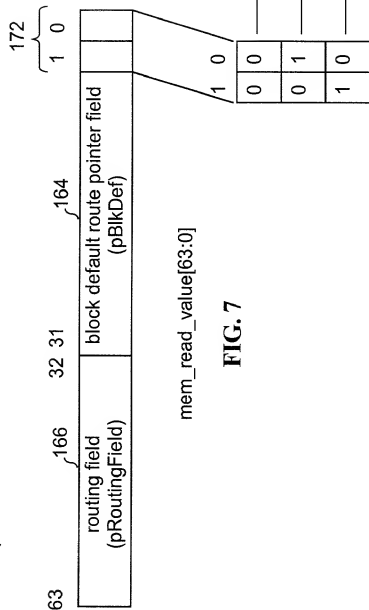
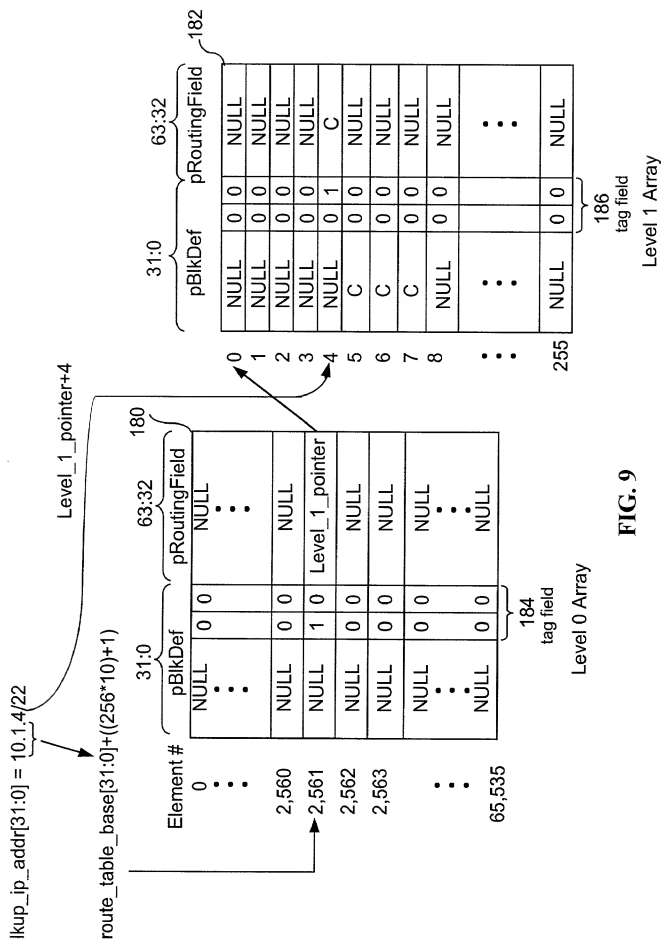


FIG. 7

FIG. 8



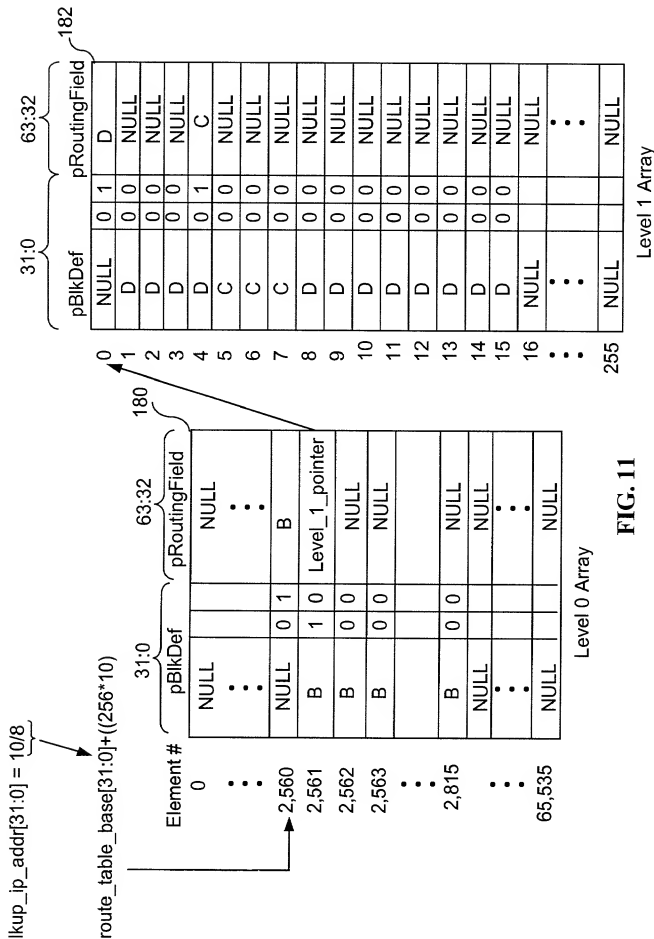
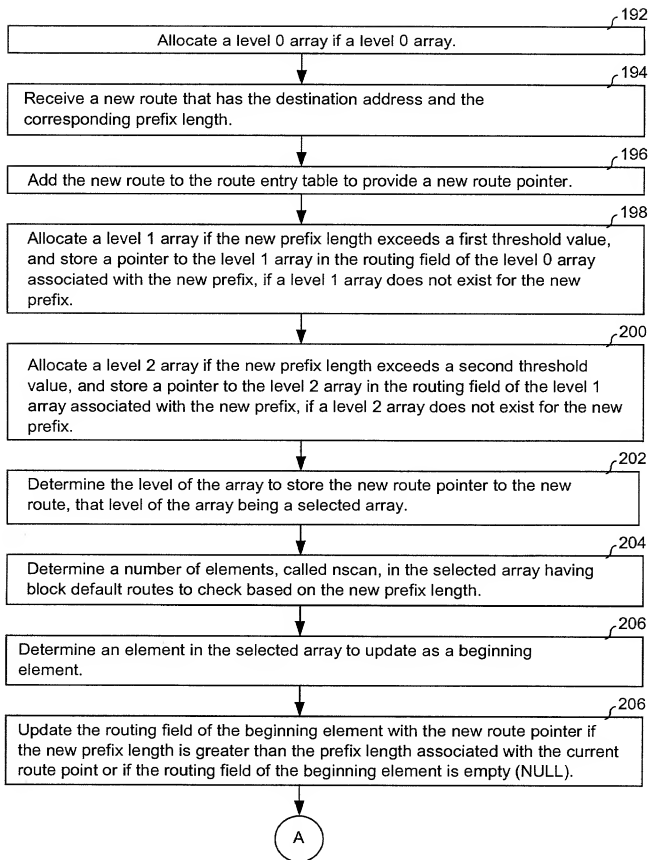
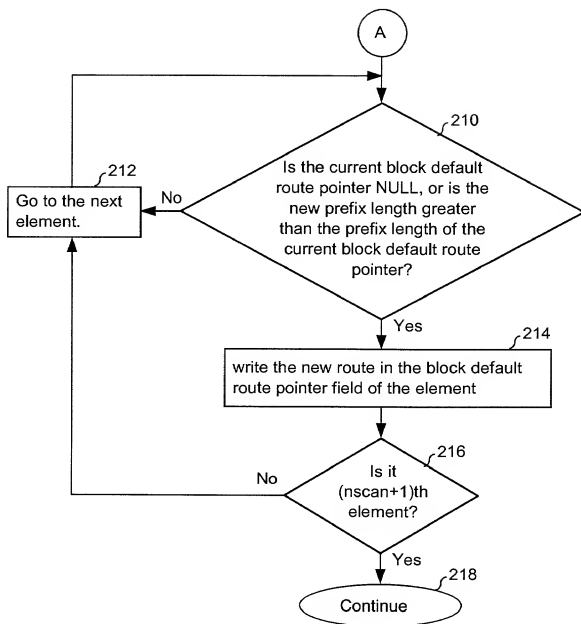


FIG. 11



Flowchart for Adding a Route

FIG. 12A



Flowchart for Adding a Route (continued)

FIG. 12B

```

addRoute(ipa, plen)
/* ipa: destination address of new route,
   plen: prefix length of new route */

```

Level 0, level 1 and level 2 arrays are allocated and next level route pointers in the level 0 and level 1 routes are updated.

```

array[] = the array to which the new route is added
begin = getIndex(ipa,plen) /* determine which array and element of that
                             array stores the new route pointer as a route */
nScan = getNscan(plen) /* determine a number of elements to scan */

/* Update the array with the new route */
array[begin].pRoutingField = pointer to the new route

/* Update the block default route */
i = begin + 1 /* i points to the next element */
While nScan-- > 0
    If plen > prefix length of the route pointed by array[i].pBlkdef then
        array[i].pBlkDef = pointer to the new route
        i = i + 1

```

Pseudo code for adding a route

FIG. 13

Memory Write Sequence for Route D Inserted
in Level 1 Array

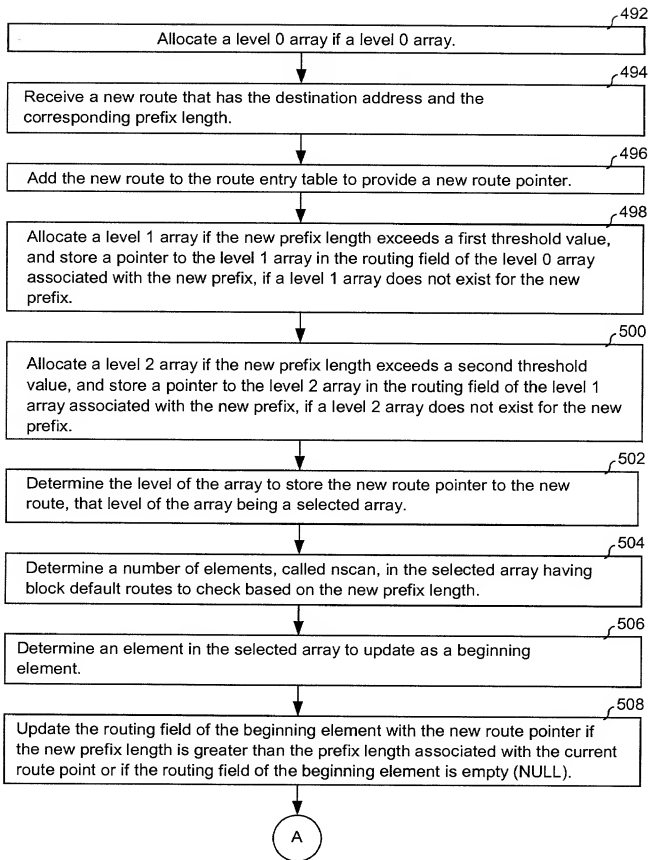
Element 0	Write D in Routing Field
Element 1	Write D in Default Route
Element 2	Write D in Default Route
Element 3	Write D in Default Route
Element 4	Write D in Default Route
Element 8	Write D in Default Route
Element 9	Write D in Default Route
Element 10	Write D in Default Route
Element 11	Write D in Default Route
⋮	⋮

FIG. 14

Memory Write Sequence for Route D Deleted
from Level 1 Array

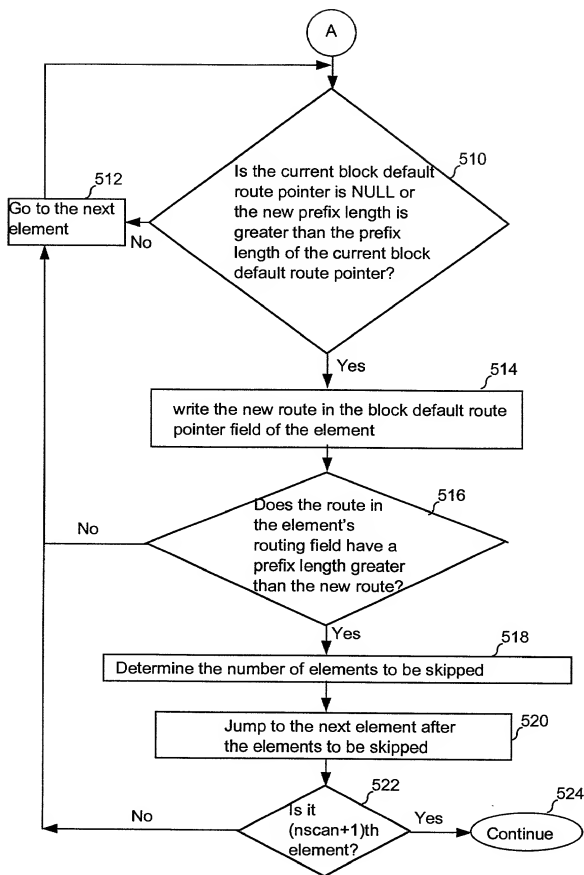
Element 0	Delete D from Routing Field
Element 1	Delete D from Default Route
Element 2	Delete D from Default Route
Element 3	Delete D from Default Route
Element 4	Delete D from Default Route
Element 8	Delete D from Default Route
Element 9	Delete D from Default Route
Element 10	Delete D from Default Route
Element 11	Delete D from Default Route
⋮	⋮

FIG. 17



Flowchart for Adding a Route with Automatic Skipping

FIG. 15A



Flowchart for Adding a Route with Automatic Skipping (continued)

FIG. 15B

```

addRoute(ipa, plen)
/* ipa: destination address of new route,
   plen: prefix length of new route */

```

Level 0, level 1 and level 2 arrays are allocated and next level route pointers in the level 0 and level 1 routes are updated.

```

array[] = the array to which the new route is added
begin = getIndex(ipa,plen) /* determine which array and element of that
                           array stores the new route pointer as a route */
nScan = getNscan(plen) /* determine a number of elements to scan */

/* Update the array with the new route */
array[begin].pRoutingField = pointer to the new route

/* Update the block default route */
i = begin + 1 /* i points to the next element */
While nScan-- > 0
    If plen > prefix length of the route pointed by array[i].pBlkDef then
        array[i].pBlkDef = pointer to the new route
        i = i + 1
If array[i].pRoutingField = Not Null then
    nSkip = getNscan (plen) + 1 /* get number of elements
                                to be skipped */
    i = i + nSkip /* jump to the next element after skipping */

```

Pseudo Code for Enhanced Route Addition with Automatic Skipping

FIG. 16

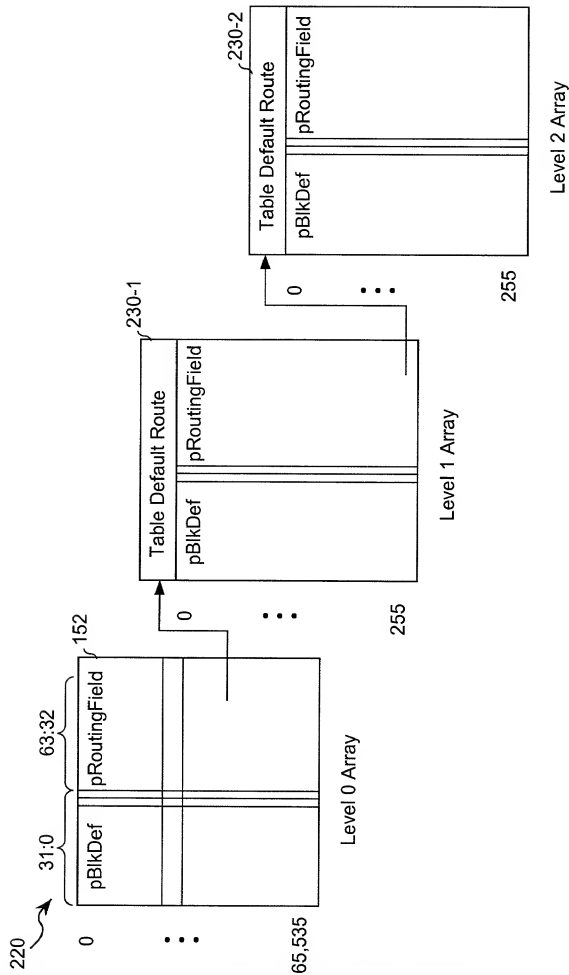


FIG. 18

240

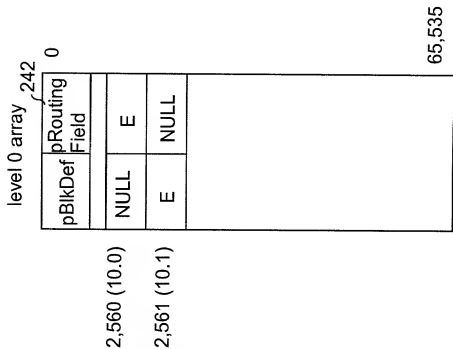


FIG. 19A

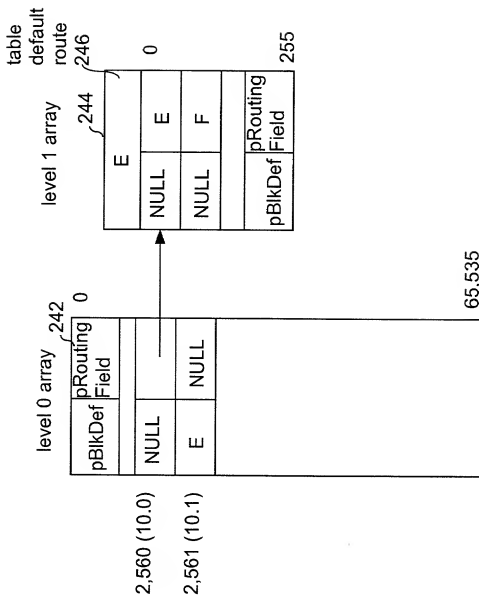
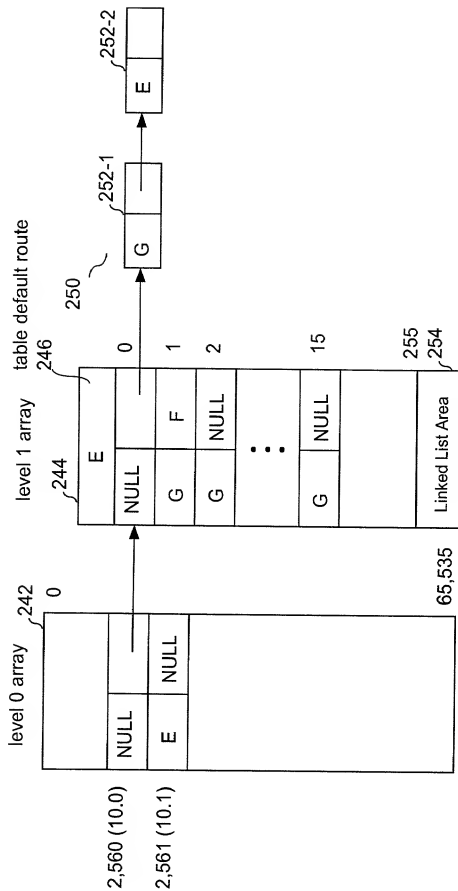


FIG. 19B



Overlapping Routes
FIG. 20

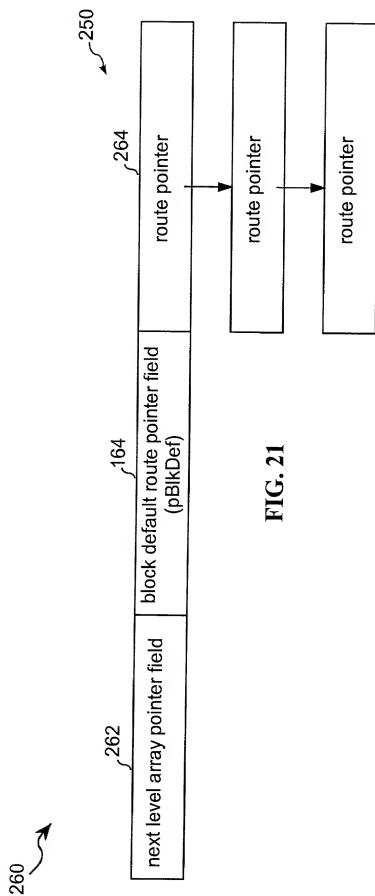


FIG. 21

Route Engine
120

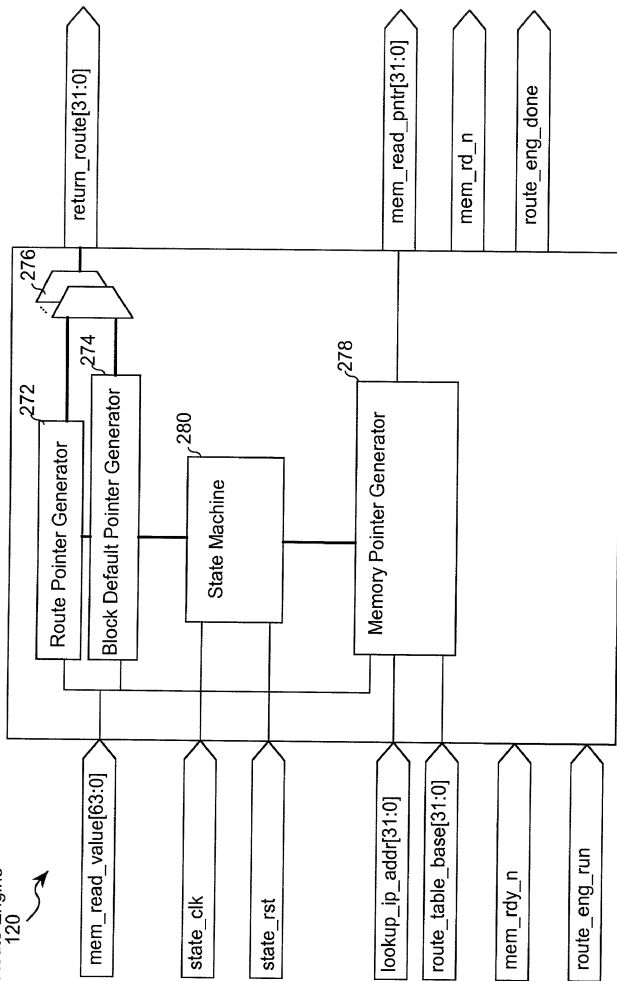


FIG. 22

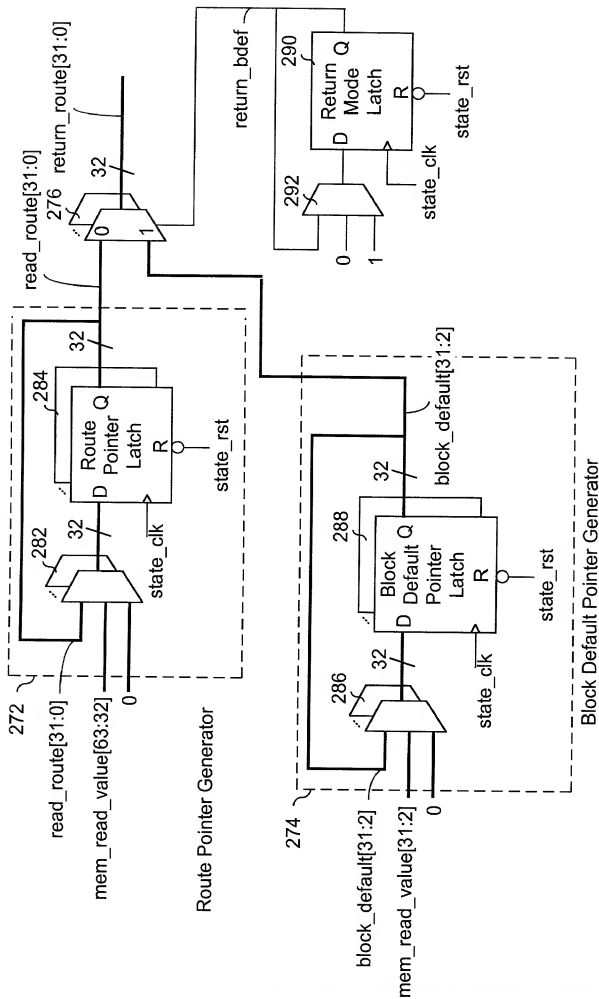


FIG. 23

Memory Pointer Generator

278

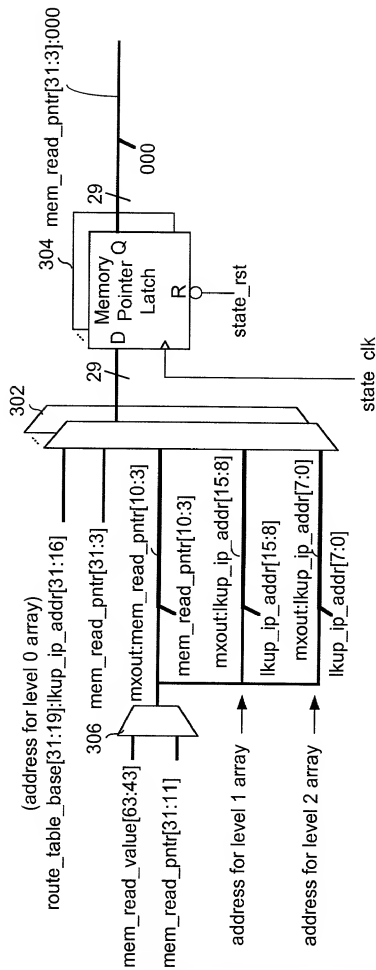


FIG. 24

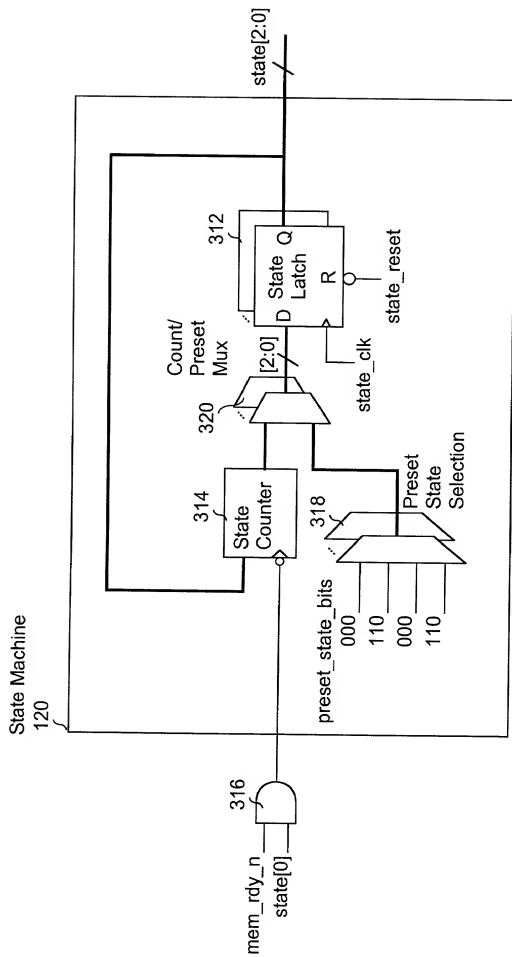


FIG. 25

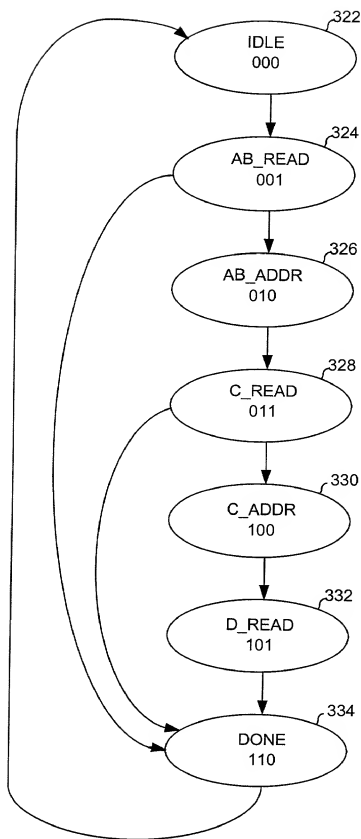


FIG. 26